White Paper

2025

Building Enterprise Generative AI Applications

A Guide to Getting Started from Booz Allen's GenAI Team

Booz Allen®

Table of Contents

Introduction

Generative artificial intelligence (GenAI) is reshaping how enterprises approach knowledge management, decision making, and user interaction. While this technology has immense potential, deploying GenAI at scale within an enterprise requires more than just model access—it demands a strategic, layered approach that aligns with business goals, data infrastructure, and governance standards.

This report presents a comprehensive framework for building enterprise-grade GenAI applications, structured around a six-layer technology stack architecture: Infrastructure, Platform, Large Language Model (LLM), Data and Data Pipeline, Capability and Agent, and User Interface (UI)/ Application. Each layer plays a critical role in ensuring scalability, security, and performance.

Key considerations include selecting the right deployment model (on-premises, cloud, or hosted application programming interfaces [APIs]), choosing and orchestrating LLMs based on task complexity and cost, and preparing high-quality data pipelines to support real-time and domain-specific use cases. We also emphasize the importance of embedding human oversight into AI workflows, rather than relying solely on autonomous agents.

Beyond architecture, our report outlines essential practices for success: implementing robust LLM operations (LLMOps) for continuous monitoring and improvement and establishing strong governance, risk, and compliance (GRC) frameworks. These practices include bias mitigation, security safeguards, and ethical guardrails to ensure responsible AI deployment.

By following this structured approach, organizations can unlock the full potential of GenAI—delivering intelligent, reliable, and ethically sound applications that drive measurable business outcomes.

The GenAI Tech Stack Architecture

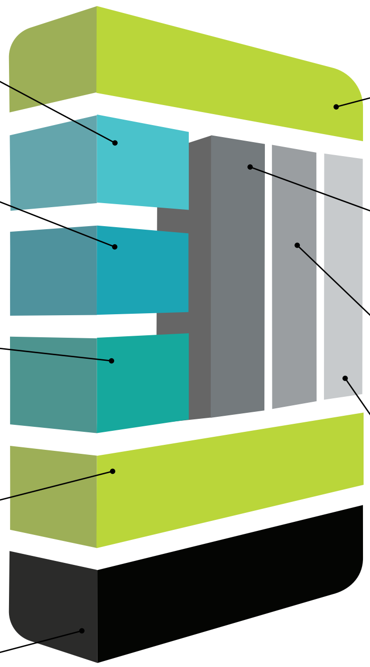
While commercial LLMs deliver impressive capabilities out of the box, these are not enterprise-ready GenAI applications, at least not in a conventional sense. Rather, a GenAI application builds upon a complex ecosystem of specialized tools and technologies and orchestrated workflows and techniques.

To begin with, it is critical to integrate AI systems with mission-specific knowledge, rules, and workflows to deliver contextually appropriate outputs for federal environments. Implementing guardrails, such as fact-checking mechanisms and context-aware validations, helps mitigate risks of hallucination and other errors, improving reliability and accuracy. Advanced security measures further enable agencies to prevent misuse and safeguard sensitive data and user privacy from external attacks.

At the same time, organizations need to ensure GenAI applications are scalable and performant enough to serve the most critical missions while being customizable and configurable enough to solve real problems and deliver real impact. This includes avoiding technology lock-in by building extensible, forward-compatible solutions. Achieving this agility requires the use of standards-based, open architectures that enable plug-and-play adoption of best-of-breed components.

As we will explore, a GenAI tech stack provides the architecture, capabilities, and operating structure needed to fill this void. The key components or layers of a GenAI tech stack that integrates engineering best practices include:

GenAI Tech Stack



Agents & Capability

Tools and techniques used to coordinate and execute an autonomous workflow.

Data and Data Pipeline

Mission-specific data used for steering or training models.

Large Language Model (LLM)

Complex algorithms and data to learn patterns and structures of language, allowing them to generate human-like text, speech, and images.

Platform

Software and tools that enable the development, deployment, and management of LLM applications.

Infrastructure

Underlying physical and virtual resources required to support the model (e.g., hardware, storage solutions, networking resources).

UI/Application

End-user software applications, including web interfaces, desktop applications, mobile apps, to command-line interfaces (CLI) through which users access and use the LLM.

Governance, Risk, and Compliance (GRC)

Interrelated disciplines used to operate GenAI systems securely, safely, and responsibly.

LLMOps

The pipelines necessary to rapidly design, develop, test, evaluate, deploy, and continuously monitor and improve generative AI solutions.

Orchestration

Workflow/function-based or agentic frameworks and communication protocols.

Figure 1. A robust GenAI tech stack consists of six functional layers enabled by three cross-cutting operational processes

Infrastructure and Platform Layers

Infrastructure refers to the physical or cloud-based resources that power data storage, processing, and AI computations. A robust infrastructure ensures systems can efficiently manage large datasets and complex computations, particularly for real-time applications where minimizing latency is crucial.

This computational infrastructure is at the foundation of any GenAI system. Enterprises must evaluate three primary deployment models:

- On-Premises (Self-Managed LLM): Dedicated hardware (e.g., Nvidia DGX, graphics processing unit [GPU] clusters) provides full control but requires high maintenance.
- Cloud-Based (Self-Managed LLM): Cloud platforms like Amazon Web Services (AWS), Microsoft Azure, and Google Cloud allow organizations to deploy models with scalable compute resources.
- Cloud-Based (Hosted API LLM): Managed services like OpenAI's GPT, Azure OpenAI, or Anthropic's Claude provide ease of use but may introduce vendor lock-in.

As detailed in Table 1, each option involves tradeoffs regarding control, scalability, cost, and ease of use. This comparison helps technical leaders evaluate the best fit for their organization's goals and resources.

The **Platform** layer includes the software and tools required to develop, train, and deploy AI models. This provides developers with the necessary environments and tooling to code, test, and refine AI algorithms, playing a vital role throughout the AI lifecycle.

Option	On-Premises (Self-Managed LLM)	Cloud-Based (Self-Managed LLM)	Cloud-Based (Hosted API LLM)
Integration Complexity	High integration complexity; requires custom development to align with existing systems.	Moderate complexity; integrations depend on cloud services and APIs.	Low complexity; APIs enable faster integration with minimal effort.
Compliance and Regulations	Ideal for highly regulated industries requiring complete control over compliance.	Cloud providers offer compliance support, but shared responsibility for data security remains.	Compliance depends on provider certifications; organizations must ensure data protection policies are met.
Time to Implementation	Longer implementation time due to hardware procurement, setup, and configuration.	Moderate setup time; requires configuration and fine-tuning of cloud infrastructure.	Fastest deployment; ready-to-use APIs with minimal configuration.
Cost Management	Predictable costs but large upfront investments.	Costs fluctuate based on compute and storage usage.	Subscription-based pricing can simplify budgeting, but costs can scale rapidly with high usage.
Resilience and Uptime	Uptime depends on in-house infrastructure and disaster recovery solutions.	Cloud providers offer high availability but require proper setup for failover.	Built-in redundancy ensures high uptime, managed by the provider.

Table 1 : Infrastructure Considerations

LLM Layer

LLMs are the core component for most enterprise GenAI applications due to their reasoning abilities and capacity to generate human-like language. Trained on massive amounts of data, LLMs are the backbone of GenAI capabilities. However, not all models have the same capabilities, training data, or performance characteristics.

For example, their usage agreements often vary, creating specific constraints. More importantly, different models have different performance competencies ranging from translation and summarization to code generation and structured data analysis. For some use cases, smaller models fine-tuned on enterprise domain-specific data can outperform larger ones. Performance metrics, compute need, and cost also play significant roles in the process of selecting the right model for a given use case.

These complexities may require the use of several models, making intelligent orchestration and routing necessary. In multi-model applications, an **orchestration layer** analyzes incoming requests and routes them to the most appropriate model for the task. Factors that may impact routing decisions include model capabilities, performance needs, data sensitivity, and cost.

Selecting the appropriate model, or models, is critical to ensuring a successful application. However, new models are being continually released with new capabilities, different context windows, improved performance, and different costs. With GenAI rapidly changing, it is crucial that the LLM layer is adaptive and responsive to change. As new models are released and architected, they can be added while existing models are replaced. Developing a culture of continual experimentation, evaluation, and improvement ensures GenAI applications remain relevant with the best models for each use case.

Data and Data Pipeline Layer

Enterprise GenAI applications rely on high-quality, well-structured data. To achieve this, data pipelines must address:

- **Preprocessing:** Data cleaning, entity extraction, relation mapping, and embedding generation.

- **Transformation:** Tokenization, vectorization, and metadata enrichment for RAG.

- **Integration:** Connection of structured (e.g., SQL databases) and unstructured (e.g., document stores) sources.

At its simplest, data preparation may involve basic cleaning and formatting, such as removing personally identifiable information (PII), irrelevant data, or incomplete data; converting data into a consistent format (e.g., standardizing text encoding); and tokenization, where text is broken into smaller, more manageable units for model processing.

Retrieval-Augmented Generation (RAG) is an AI framework that enhances large language models by retrieving relevant external documents or data in real time. It combines this retrieved context with generative capabilities to produce more accurate, grounded, and up-to-date responses, especially useful for enterprise search, question answering, and decision support.

For more complex tasks, preprocessing becomes a multi-step, intricate process involving advanced techniques:

- **Information Extraction:** Automatically identifies and retrieves structured data or entities (e.g., names, addresses, sentiments) from unstructured content to make the information usable for analysis or automation.

- **Relation Extraction:** Identifies relationships between extracted entities, such as linking individuals to organizations or events.

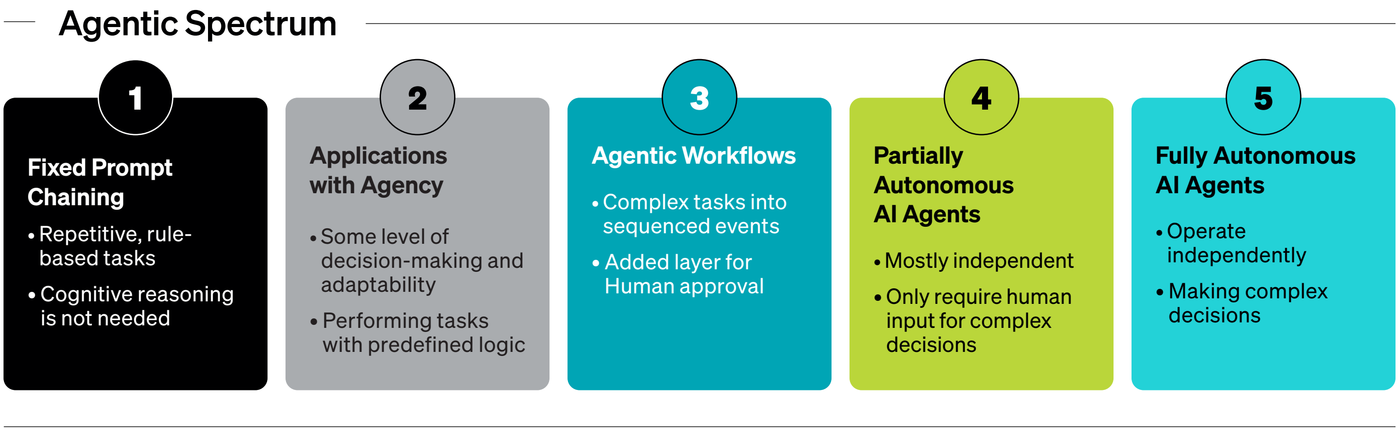
- **Entity Linking:** Associates identified entities with real-world references or external databases to provide context and enhance understanding.

Data is another foundation of all AI systems, and this holds particularly true for GenAI. High-quality, robust, diverse, and complete data is essential to ensure the accurate, relevant, and coherent outputs expected from these sophisticated models. The role of data is critical across the entire lifecycle: it forms the basis for (1) foundational model training, provides the specific knowledge needed for (2) fine-tuning to enterprise contexts, and enables (3) real-time model steering during inference, particularly through techniques like RAG. Effective data strategy and meticulous preparation are therefore prerequisites for building powerful and reliable enterprise GenAI applications. Furthermore, how data is organized—its structure and hierarchy—directly impacts system complexity, computational demands, and, ultimately, the balance among processing speed, accuracy, and resource utilization.

Agent and Capability Layer

While many users interact with GenAI through simple chatbot interfaces—asking questions and receiving responses—this model only scratches the surface of what's possible. To move beyond passive interaction and enable systems to take meaningful action on behalf of users, **AI agents** are essential. These agents allow applications to operate with varying degrees of autonomy, from offering intelligent suggestions to executing complex workflows independently.

As organizations develop these **agentic AI systems**, they can calibrate the level of agency and autonomy granted to these agents. This flexibility is critical for optimizing performance while managing operational and ethical risks. By tuning how much control an agent has—whether it simply recommends actions or takes them automatically—enterprises can strike the right balance between automation and oversight.



Agentic Spectrum

1 Fixed Prompt Chaining
- Repetitive, rule-based tasks
- Cognitive reasoning is not needed

2 Applications with Agency
- Some level of decision-making and adaptability
- Performing tasks with predefined logic

3 Agentic Workflows
- Complex tasks into sequenced events
- Added layer for Human approval

4 Partially Autonomous AI Agents
- Mostly independent
- Only require human input for complex decisions

5 Fully Autonomous AI Agents
- Operate independently
- Making complex decisions

For example, applications with incremental agency can enhance the user experience through adaptive support and intelligent suggestions without overwhelming autonomy. This fosters a collaborative environment where human oversight and AI capabilities complement each other seamlessly, enhancing effectiveness and efficiency.

This is important, as current AI agents often underperform humans in real-world scenarios. For instance, the Claude AI Agent Computer Interface (ACI) achieves only 14 percent of human-level performance, demonstrating the necessity of a spectrum of agency. The top-performing models resolve a limited percentage of tasks while incurring high computational costs. These limitations highlight the importance of human oversight embedded within AI systems to balance reliance on agents against projected cost and time savings.

AI agents are designed to autonomously perform tasks by leveraging LLMs to decompose complex tasks into manageable sub-tasks. However, they struggle with long-horizon tasks, requiring human intervention as checkpoints before executing tasks. This human-in-the-loop (HITL) approach addresses concerns regarding autonomy, positioning humans as valuable guides when agents encounter limitations.

Related capabilities, such as prompt templates, standardized agent interfaces (e.g., Model Context Protocol), and in-process memory, can be used to further enhance reasoning. These capabilities are typically modular and composable—agents can be equipped with different sets depending on the domain or task—and can be used to develop logic for sequencing actions and adapting to new inputs. For example, a compliance-checking agent might use document parsing, policy retrieval, and summarization capabilities.

[*Editor's note: The article* "The Age of Agentic AI" *in the current issue of Velocity magazine provides more insight on this topic.*]

UI/Application Layer

Once potential applications are identified, the next step is to determine the most effective way for users (or other systems) to interact with the GenAI capability. The choice of interaction paradigm significantly impacts usability, workflow integration, and overall effectiveness:

- **Conversational Interfaces (Chatbots/Assistants):** Ideal for interactive tasks like Q&A, brainstorming, customer support, or step-by-step guidance. They allow for natural language interaction but require careful design to handle ambiguity and manage conversation flow.

- **Autonomous Backend Processes:** Suitable for executing multi-step tasks or workflows with a degree of independence based on a given goal. These are more complex to build and manage, requiring robust planning, error handling, and safety mechanisms. They are best suited for well-defined, automatable processes where proactive execution is beneficial.
- **Structured Pipelines/API Integrations:** GenAI acts as a component within a larger automated workflow, often invoked via API calls. Examples include automatically summarizing uploaded documents, generating product descriptions based on structured data inputs, or classifying incoming support tickets. Human interaction might be minimal or indirect.
- **Augmenting Existing Applications:** Embedding GenAI features directly into users' current tools (e.g., a "summarize" button in an email client, a "draft response" feature in a customer relationship management system, code suggestions in an integrated development environment). This minimizes context switching and friction, promoting adoption for task-specific assistance.
- **Standalone Tools:** Creating a dedicated application for a specific, complex GenAI function (e.g., an advanced research analysis tool, a specialized content generation platform). This is appropriate when the task is distinct enough or requires a specialized interface not easily integrated into existing software.

The optimal paradigm depends on the nature of the task, the target users' technical proficiency, the required level of control and oversight, and the desired level of integration with existing systems.

GenAI Tech Stack Practices

Three sets of practices are critical to enabling effective, reliable, and trustworthy performance from enterprise GenAI systems: system-level orchestration frameworks and communication protocols that can stitch horizontal build components together, LLM operations (LLMOps) stemming from machine learning operations (MLOps) best practices, and critical incorporation of AI governance, risk, and compliance (GRC). They operate effectively as vertical layers within the GenAI tech stack, delivering cross-cutting capabilities.

LLMOps provides the technical foundation for deploying and maintaining GenAI models in production, with a focus on real-time performance monitoring, output quality evaluation, and continuous improvement through feedback loops and retraining. In parallel, AI GRC addresses the broader organizational responsibilities of deploying GenAI responsibly—ensuring systems are secure, ethically aligned, and compliant with regulatory standards.

Together, these frameworks enable enterprises to operationalize GenAI at scale while mitigating risks such as bias, hallucinations, security vulnerabilities, and model drift. By embedding both LLMOps and GRC into the GenAI lifecycle, organizations can ensure their AI applications remain effective, accountable, and aligned with business and societal expectations.

LLMOps: Monitoring, Evaluation, and Continuous Improvement

Deploying GenAI effectively requires continuous oversight and adaptation, managed through LLMOps. LLMOps provides the essential framework for automating and managing the GenAI model lifecycle in production, ensuring models remain effective, reliable, and updated. Unlike traditional AI MLOps focused on structured data and clear accuracy metrics, GenAI LLMOps must handle large unstructured datasets, non-deterministic outputs, and nuanced quality assessments, demanding sophisticated monitoring and feedback loops.

Real-Time Performance Monitoring

Maintaining operational health is crucial. This involves tracking key metrics in real time: latency (response time), throughput (requests handled), error rates, and resource utilization (compute, memory, token consumption for cost management). Using monitoring tools and automated alerts allows for proactive issue detection, resource optimization, and a positive user experience.

Output Quality and Drift Monitoring
Critically for GenAI, monitoring must extend beyond operational metrics to assess the quality and consistency of its non-deterministic outputs. This includes evaluating relevance, coherence, factual accuracy (hallucinations), and safety (bias, harmful content). As data patterns shift, models can experience drift, degrading performance. Effective monitoring combines automated techniques (like semantic similarity, toxicity scores, or custom classifiers) with essential HITL review processes for subjective or high-risk evaluations. Statistical drift detection methods and structured logging of inputs/outputs are vital for identifying and diagnosing quality degradation.

Feedback Loops and Retraining Strategy
Continuous improvement hinges on robust feedback mechanisms and update strategies. Feedback, gathered explicitly (user ratings) or implicitly (user interactions), informs model refinement. Performance degradation detected via monitoring or accumulated feedback triggers updates, which may involve retraining, fine-tuning on new data, or iterative prompt refinement. These updates should be managed through automated continuous integration/continuous delivery pipelines incorporating rigorous testing, version control for models and data, and safe deployment strategies like canary releases to ensure reliability.

Integrating these monitoring, evaluation, and feedback components within a tailored LLMOps framework enables enterprises to maintain high-performing, reliable, and continuously improving GenAI applications.

AI GRC

Successfully integrating GenAI applications into enterprise operations necessitates a comprehensive framework for GRC. GenAI technologies, with their probabilistic nature and potential to generate novel content, introduce unique and complex challenges beyond those of traditional software, touching upon bias, security, ethical considerations, and operational reliability. Establishing clear GRC practices is not merely a check-the-box exercise for regulatory adherence; it is fundamental to building and maintaining trust with users and stakeholders, mitigating significant reputational and financial risks, and ensuring that these powerful technologies deliver tangible business value responsibly and sustainably.

Bias Detection and Mitigation
A primary concern with GenAI models, often trained on vast, uncurated internet datasets, is their potential to inherit and amplify societal biases. This can manifest as discriminatory outputs, reinforcement of stereotypes, or unfair representation across demographic groups, posing serious ethical, reputational, and legal risks. Bias can stem from various sources, including skewed training data, inherent model architecture biases, the fine-tuning process, or even the prompts used during interaction.

Proactive detection involves using specialized tools to audit datasets for representation gaps, employing quantitative fairness metrics (e.g., demographic parity, equalized odds) to evaluate model outputs, and incorporating rigorous human review, especially for sensitive applications. Mitigation strategies are multifaceted. They range from curating more balanced datasets and employing algorithmic bias reduction techniques during or after training to ensuring meticulous prompt engineering and implementing filters on model outputs. Continuous monitoring post-deployment is essential to identify and address biases that may emerge over time.

Security Vulnerability Management
GenAI systems introduce novel attack surfaces alongside traditional software vulnerabilities. Exploitation of these can lead to unauthorized data access, manipulation of outputs for malicious purposes, system misuse, or erosion of model integrity. Key GenAI-specific threats demand targeted defenses:

- **Prompt Injection:** Crafting inputs to bypass safety controls, coerce the model into revealing sensitive information, or execute unintended actions.
- **Data Poisoning:** Introducing malicious examples into training or fine-tuning data to degrade performance, embed biases, or create hidden backdoors.
- **Model Inversion:** Querying the model strategically to infer potentially sensitive information about its original training data.
- **Denial of Service (DoS):** Overloading the system with resource-intensive prompts or exploiting computational bottlenecks.

Mitigation requires a layered security posture encompassing robust input validation and sanitization, output filtering to prevent data leakage or harmful content generation, stringent access controls, continuous monitoring for anomalous activity, and specialized security testing, including adversarial "red-teaming" exercises.

Responsible AI and Ethical Guardrails

Deploying GenAI responsibly extends beyond technical security and bias checks; it requires embedding ethical principles into the AI lifecycle. Responsible AI (RAI) aims to ensure AI systems are designed, developed, and deployed to empower users, benefit the organization, and impact society fairly, thereby fostering trust and enabling confident AI scaling. Central pillars include:

- **Transparency**: Clearly documenting data sources, model capabilities and limitations, training methodologies, and intended use cases. While full model introspection remains challenging, procedural transparency is key.
- **Explainability (Interpretability):** Providing insights into why a model produced a specific output, even if full causality is elusive. Techniques like highlighting influential input features or citing retrieved evidence (in RAG systems) aid debugging and trust.
- **Accountability:** Establishing clear ownership and oversight for AI systems, including processes for auditing, impact assessments, and mechanisms for redress when errors or harm occur.
- **Content Safety and Fairness:** Implementing technical controls (filters, classifiers) and robust usage policies to prevent the generation of harmful, illegal, or unfairly biased content, coupled with active monitoring for compliance. Operationalizing these principles necessitates cross-functional collaboration involving legal, ethical, technical, and business teams.

Testing, Validation, and Reliability Assurance

Ensuring enterprise GenAI applications are reliable, are accurate, and perform consistently is crucial but complex due to their generative nature. Testing must go beyond standard software quality assurance to evaluate probabilistic outputs:

- **Functional Correctness and Output Quality:** Verifying performance against specifications while rigorously assessing output relevance, coherence, factual accuracy (hallucination detection), adherence to brand voice, and safety. This often requires combining automated metrics (e.g., ROUGE, BLEU, toxicity scores) with structured human evaluation.
- **Performance and Scalability:** Assessing latency, throughput, and resource usage under realistic load conditions.
- **Robustness and Edge Cases:** Testing system responses to unexpected, ambiguous, or adversarial inputs to identify failure modes.
- **Reliability and Consistency:** Continuously monitoring performance over time to detect degradation or drift. A comprehensive testing strategy, integrating automated and human checks throughout development and post-deployment, is essential for building dependable GenAI solutions.

Conclusion

Building enterprise GenAI applications demands a structured approach to architecture, data integration, model selection, and governance. By aligning each layer of the GenAI technology stack with business objectives, organizations can deploy scalable, secure, and high-performing AI applications. Thoughtful design decisions at every level ensure not only technological success but also responsible and ethical AI deployment for real-world impact.

Before designing any GenAI application, organizations should understand the business value and return on investment (ROI) expected from the implementation. Each decision in developing the application and selecting the right stack should be made with maximizing ROI in mind. It is critical to understand how success will be measured and to identify, collect, and analyze key metrics that will guide the evaluation and assessment.

A key decision that should be made early in the development process is whether GenAI is even the most appropriate tool. Depending on the task and how success is measured, traditional AI/ML approaches or non-AI approaches may be the most effective in delivering the desired ROI. Regardless of approach, another key consideration is whether an application will replace human effort, augment human work, or implement an automated workflow. In any of these cases, GenAI capabilities must be mapped to the business process, whether it is content creation, data analysis, information extraction, or any number of tasks where GenAI may assist.

Focusing on specific, high-value tasks within existing workflows provides a clear target for application design and helps ensure the resulting solution addresses a genuine organizational need. Keeping this in mind, it is critical to also understand how the application fits into the larger enterprise ecosystem. ROI, business workflows, and tool selection will all be impacted by enterprise considerations such as enterprise observability and security requirements, existing vendor/service provider agreements, data sensitivity, organization risk factors, and regulatory requirements.

About Booz Allen

Booz Allen is the advanced technology company delivering outcomes with speed for America's most critical defense, civil, and national security priorities. We build technology solutions using AI, cyber, and other cutting-edge technologies to advance and protect the nation and its citizens. By focusing on outcomes, we enable our people, clients, and their missions to succeed—accelerating the nation to realize our purpose: Empower People to Change the World®.

BoozAllen.com

Booz
Allen®